The Complexity of Valued Constraint Satisfaction Problems in a Nutshell

Johan Thapper

The Complexity of Valued Constraint Satisfaction Problems in a Nutshell

Johan Thapper[1] *

[1] École Polytechnique, Laboratoire d'Informatique (LIX), 91128 Palaiseau, France
thapper@lix.polytechnique.fr

Abstract

The valued constraint satisfaction problem was introduced by Schiex et al. [23] as a unifying framework for studying constraint programming with soft constraints. A systematic worst-case complexity theoretical investigation of this problem was initiated by Cohen et al. [4], building on ideas from the successful classification programme for the ordinary constraint satisfaction problem. In addition to the decision problem for constraint satisfaction, this framework also captures problems as varied as Max CSP and integer programming with bounded domains.

This paper is intended to give a quick introduction to the questions, the main results, and the current state of the complexity classification of valued constraint satisfaction problems. Two special cases are looked at in some detail : the classification for the Boolean domain and the less well-understood case of Max CSP. Some recent results for general constraint languages are also reviewed, as well as the connection to the very active study of approximation algorithms for Max CSP.

1 Introduction

The valued constraint satisfaction problem was introduced by Schiex et al. [23] as a unifying framework for studying constraint programming with soft constraints. It has also proved to be a convenient framework for studying the complexity of various optimisation variations of constraint satisfaction problems. This paper gives a quick introduction to the type of questions studied in this area by reviewing the main results for the Boolean domain and for Max CSP. The focus is on the *constraint language* parameterisation, but other complexity questions have also been considered, see for example [6].

*The author has received funding from the ERC FP7/2007-2013 Grant Agreement no. 257039.

In the original definition by Schiex et al. [23], a VCSP is defined over a *valuation structure* ; a totally ordered set together with an *aggregation operator* satisfying certain basic properties. Here we will exclusively consider the valuation structure $\overline{\mathbb{Q}}_{\geq 0}$; the extended non-negative rational numbers with the natural aggregation operator $+$, where $x + \infty = \infty$ for all x.

Definition 1.1 *A VCSP-instance consists of a triple (V, D, C), where*
- *V is a finite set of* variables *;*
- *D is a finite set of* domain elements *; and*
- *C is a finite set of constraints $c = \langle \bar{x}, f \rangle$, where $\bar{x}$,* the scope, *is a tuple of variables from V and f is a function from $D^{|\bar{x}|}$ to $\overline{\mathbb{Q}}_{\geq 0}$.*

A constraint with a function f is called *soft* if f does not take any infinite value. It is called *crisp* if f takes values in $\{0, \infty\}$. An *assignment* to a VCSP-instance $\mathcal{I}$ is a function $\sigma : V \to D$. The *measure*, $m_{\mathcal{I}}(\sigma)$, of σ is the total aggregated cost of the constraints of $\mathcal{I}$,

$$m_{\mathcal{I}}(\sigma) := \textstyle\sum_{\langle \bar{x}, f \rangle \in C} f(\sigma(\bar{x})),$$

where σ is applied component-wise to $\bar{x}$. The measure is sometimes also denoted by $Cost_{\mathcal{I}}(\sigma)$. The objective is to *minimise* the measure over all assignments.

A *(valued) constraint language* $\mathcal{F}$ is a set of functions $f : D^k \to \overline{\mathbb{Q}}_{\geq 0}$. The problem VCSP($\mathcal{F}$) is the set of VCSP-instances in which the cost functions come from $\mathcal{F}$. If there exists an algorithm that solves VCSP($\mathcal{F}$) in polynomial time, then $\mathcal{F}$ is said to be *tractable*. If there is a polynomial-time reduction to VCSP($\mathcal{F}$) from some NP-hard problem, $\mathcal{F}$ is said to be NP-hard.

Example 1 (CSP) *A standard constraint satisfaction problem is given by a set of relations applied to tuples of variables : $R_1(\bar{x}_1), \ldots, R_m(\bar{x}_m)$. This can be* *expressed as a VCSP using the following translation : For each k-ary R_i, define the function $f_{R_i} : D^k \to \overline{\mathbb{Q}}_{\geq 0}$ by $f_{R_i}(\bar{t}) = 0$ if $\bar{t} \in R_i$ and $f_{R_i}(\bar{t}) = \infty$ otherwise. The VCSP instance is then given by the constraints $\langle \bar{x}_i, f_{R_i} \rangle$ for $i \leq m$. An assignment $\sigma : V \to D$ satisfies the original instance iff $\sum_i f_{R_i}(\sigma(\bar{x}_i)) = 0$.*

Example 2 (Max CSP) *One of the most well-studied optimisation variations of CSP is the Max CSP problem : Rather than asking whether an instance is satisfiable, one wants to maximise the number of satisfied constraints. Although this is a maximisation problem, it can for some purposes be modelled as a VCSP*[1]*. In this case the translation takes a relation R to the function f_R such that $f_R(\bar{t}) = 0$ if $\bar{t} \in R$ and $f_R(\bar{t}) = 1$ otherwise. Hence, the degree to which non-satisfying assignments are penalised depends on the number of unsatisfied constraints.*

Example 3 (Min Ones) *Given a CSP-instance over the Boolean domain, the problem Min Ones is to either decide that the instance is unsatisfiable, or to return a satisfying assignment with as few variables as possible set to the value 1 (true). The weighted version of this problem can be expressed as a VCSP by taking the functions used in Example 1 for expressing the crisp constraints together with the unary function $f(t) = t$ for $t = 0, 1$.*

The last example can be seen as the result of adding a linear objective function to a CSP. A generalisation of this idea to larger domains leads to the definition of *the maximum solution problem* [12, 14, 15]. The generalised problem can also be expressed as a VCSP. It models a variety of problems including integer programming over bounded domains.

2 The Boolean Case

A relation is 0*-valid* (resp. *1-valid*) if it contains the all-0 (resp. all-1) tuple. A Boolean relation is 2*-monotone* if it can be written on the form $R(x_1, \ldots, x_p, y_1, \ldots, y_q) \equiv (x_1 \wedge \cdots \wedge x_p) \vee (\neg y_1 \wedge \cdots \wedge \neg y_q)$. A set of Boolean relations Γ is said to be 0-valid (1-valid, 2-monotone) if all its relations are 0-valid (1-valid, 2-monotone).

The following is good example of the type of result one encounters in this area :

Theorem 2.1 (Creignou [7]) *Let Γ be a set of Boolean relations. Then Max CSP(Γ) can be solved in polynomial time if Γ is* 0-valid, 1-valid, *or if Γ is* 2-monotone. *Otherwise the problem is* NP*-hard.*

1. The naturally corresponding minimisation problem is equivalent with respect to exact optimisation, but *not* with respect to approximation.

That is, for some type of cost functions (finite-valued, $\{0, 1\}$-valued, etc.,) one looks for a classification of the complexity of VCSP into a small number of complexity classes. The complexity classes vary depending on the tools available. For this reason, some classifications are made up to approximation preserving reductions, while others settle for distinguishing between tractability and NP-hardness.

Khanna et al. [16] gives classifications for the Boolean domain cases of Max CSP, Min CSP, Max Ones, and Min Ones, and their weighted counterparts (see also [8]). Their main tool is called *strict implementations*, a type of gadget that preserves certain approximation properties.

Cohen et al. [4] initiated the systematic study of the complexity of VCSPs. They introduced several concepts inspired by the study of ordinary CSPs. In place of strict implementations they define *expressibility* modelled after *pp-definability*. This provides more flexibility but the reductions involved are now polynomial-time many-one reductions and do not allow the distinction between various approximation classes.

They also introduce *multimorphisms* ; a notion roughly corresponding to *polymorphisms* of relational structures used in the study of ordinary CSPs.

Let $f : D^m \to \overline{\mathbb{Q}}_{\geq 0}$ be a cost function. Let $\sqcap, \sqcup : D^2 \to D$ be two binary operations on D. Then $\langle \sqcap, \sqcup \rangle$ is called *a (binary) multimorphism of* f if, for any two m-tuples, a and b,

$$f(a \sqcap b) + f(a \sqcup b) \leq f(a) + f(b),$$

where $\sqcap$ and $\sqcup$ are applied component-wise. Multimorphisms of higher arity are defined analogously. If $\langle \sqcap, \sqcup \rangle$ is a multimorphism of every cost function in a valued constraint language $\mathcal{F}$, then $\langle \sqcap, \sqcup \rangle$ is said to be *a multimorphism of* $\mathcal{F}$.

Many known classes of tractable problems and polynomial-time algorithms can be directly linked to the fact that a valued constraint language has some particular multimorphism.

Example 4 *Crisp so-called max-closed languages are known to be solvable by arc-consistency from the study of standard CSPs. These are generalised by valued constraint languages with the multimorphism $\langle \max, \max \rangle$. All such languages are tractable.*

Example 5 *The condition for tractability in Theorem 2.1 can be reformulated in terms of multimorphisms : R is 0-valid (1-valid) iff $f_R(0, \ldots, 0) = 0$ ($f_R(1, \ldots, 1) = 0$) iff f_R has the unary multimorphism $\langle 0 \rangle$ ($\langle 1 \rangle$), where 0 and 1 denotes the constant unary functions. Furthermore, a relation R is 2-monotone iff $1 - f_R$ has the multimorphism $\langle \min, \max \rangle$.*

Armed with expressibility and multimorphisms, Cohen et al. classify *all* tractable valued constraint languages on the Boolean domain into 8 cases based on the multimorphisms they possess. These are given by :

- $\langle 0 \rangle$, $\langle 1 \rangle$;
- $\langle \min, \max \rangle$, $\langle \min, \min \rangle$, $\langle \max, \max \rangle$; and
- $\langle \text{mj}, \text{mj}, \text{mj} \rangle$, $\langle \text{mn}, \text{mn}, \text{mn} \rangle$, $\langle \text{mj}, \text{mj}, \text{mn} \rangle$,

where mj (mn) denotes the *majority* (*minority*) operation. All other valued constraint languages on the Boolean domain are NP-hard.

Note that this classification contains Schaefer's classification of Boolean CSPs [22], the Boolean Max CSP, Min CSP, Max Ones, and Min Ones problems considered by Creignou and Khanna et al., as well as problems with mixed cost functions. On the other hand, due to the nature of the reductions involved, this classification does not reproduce the various approximation classes distinguished in [16].

3 Max CSP

The two first conditions in Theorem 2.1 seem trivial : If Γ is 0-valid, then *any* instance of Max CSP(Γ) can be mapped to an equivalent instance of a language on a domain with a single element. Informally Γ is a called *core* if it cannot be reduced to a language on a smaller domain in this simple fashion.

Theorem 2.1 gives a classification for Max CSP on a Boolean domain. The three-element domain case was classified in [11]. The case when Γ contains all unary relations was classified in [9]. The proofs of these results use cleverly conducted computer-aided searches in conjunction with the strict implementations to obtain dichotomies between tractable and APX-hard constraint languages.

Let $\mathcal{F}$ be a set of $\{0, 1\}$-valued functions. All of the mentioned results for Max CSP (including Creignou's original result for the Boolean domain) can be stated for VCSP on the following form :

Assuming that $\mathcal{F}$ is a core, $\mathcal{F}$ is tractable iff it has the multimorphism $\langle \min, \max \rangle$ for some order on the domain of $\mathcal{F}$. Otherwise $\mathcal{F}$ is NP*-hard.*

Given a total order on the domain, a cost function f with a multimorphism $\langle \min, \max \rangle$ is called *submodular (on the fixed order)*, where min and max are taken with respect to the order. Submodular functions appear as an important class of tractable valued constraint languages for minimisation.

More generally, one can define submodularity over an arbitrary *lattice* on the domain. The meet and join of the lattice then become the two operations of a multimorphism. The realisation that this defines important tractable subclasses of Max CSP was made in [3], where it was also conjectured that, the only source of tractability for a core $\mathcal{F}$ is submodularity on a lattice. This conjecture was disproved in [13] where a classification for the four-element domain case showed that there are tractable $\{0, 1\}$-valued constraint languages that are not submodular with respect to any lattice.

Tractability for more general soft constraint languages based on generalisations of submodularity are considered in [2, 24].

4 Recent Developments

While multimorphisms (and various other concepts mimicking the universal-algebraic study of CSPs) have been around for a while, a completely satisfactory theory has been lacking. This is remedied by the introduction of *weighted polymorphisms* in [5], which are shown to completely determine the complexity of a general valued constraint language. The paper also defines a *Galois connection* between valued constraint languages and sets of weighted polymorphisms.

Another recent result is the classification of valued constraint languages containing all (soft) unary functions [18]. Interestingly, this result does *not* use the advanced algebraic machinery of weighted polymorphisms, but is instead based on a graph-representation of partial multimorphisms.

Parallel to the development of algebraic tools for dealing with VCSP, a different community has made immense progress on the approximability of Max CSP and other CSP-related optimisation problems. Goemans and Williamson [10] introduced rounding of semidefinite programming (SDP) relaxations as a basis for approximation algorithms. Their approximation algorithm for Max cut (Max CSP($\{\neq\}$) on a Boolean domain) achieves a constant approximation ratio of 0.87856, beating the trivial ratio 0.5 (achievable by taking a random assignment) that was up until then the best known. The Goemans and Williamson approximation ratio for Max cut has been matched by an upper bound, i.e., a hardness-result showing that, under the *unique games conjecture (UGC)* [17], the ratio obtained by the algorithm is the best possible.

Building on these ideas, Raghavendra [20] eventually managed to develop SDP-based algorithms for *every* Max CSP-problem, with constant approximation ratios that are *optimal*, provided that the UGC holds. For a tractable constraint language, Raghavendra's algorithm should achieve a ratio of 1 and thereby provide the positive part of a complete classification for Max CSP. There are however various technical complications involved. For one, the SDP relaxations can only be solved optimally up to an (arbitrarily small) constant. Furthermore, it is not clear how to determine the ratio achieved by an algorithm given a fixed constraint languages. Raghavendra and Steurer [21] show how one may in principle approximate (with any desired precision) the ratio for an arbitrarily given constraint language, but their algorithm is doubly exponential in the domain size.

Raghavendra's result on SDP relaxations, recent work on LP-relaxations [19,24], and on *robust approximation* [1, 19] seem to be bringing closer together the interests of the two communities working on, respectively, the approximation of CSPs and the classification projects of CSPs and VCSPs.

Références

[1] Libor Barto and Marcin Kozik. Robust Satisfiability of Constraint Satisfaction Problems. In *Proceedings of STOC-2012*, 2012.

[2] David Cohen, Martin Cooper, and Peter Jeavons. Generalising submodularity and Horn clauses : Tractable optimization problems defined by tournament pair multimorphisms. *Theor. Comput. Sci.*, 401(1-3) :36–51, 2008.

[3] David Cohen, Martin Cooper, Peter Jeavons, and Andrei Krokhin. Supermodular functions and the complexity of MAX CSP. *Discrete Appl. Math.*, 149(1-3) :53–72, 2005.

[4] David Cohen, Martin Cooper, Peter Jeavons, and Andrei Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11) :983–1016, 2006.

[5] David Cohen, Páidí Creed, Peter Jeavons, and Stanislav Živný. An algebraic theory of complexity for valued constraints : Establishing a galois connection. In *Proceedings of MFCS-2011*, pages 231–242, 2011.

[6] Martin C. Cooper and Stanislav Živný Tractable triangles. In *Proceedings of CP-2011*, pages 195–209, 2011.

[7] Nadia Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *J. Comput. Syst. Sci.*, 51(3) :511–522, 1995.

[8] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of boolean constraint satisfaction problems.* SIAM, Philadelphia, PA, USA, 2001.

[9] Vladimir Deineko, Peter Jonsson, Mikael Klasson, and Andrei Krokhin. The approximability of MAX CSP with fixed-value constraints. *J. ACM*, 55(4) :1–37, 2008.

[10] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42 :1115–1145, 1995.

[11] Peter Jonsson, Mikael Klasson, and Andrei Krokhin. The approximability of three-valued MAX CSP. *SIAM J. Comput.*, 35(6) :1329–1349, 2006.

[12] Peter Jonsson, Fredrik Kuivinen, and Gustav Nordh. MAX ONES generalized to larger domains. *SIAM J. Comput.*, 38(1) :329–365, 2008.

[13] Peter Jonsson, Fredrik Kuivinen, and Johan Thapper. Min CSP on four elements : Moving beyond submodularity. In *Proceedings of CP-2011*, pages 438–453, 2011.

[14] Peter Jonsson and Gustav Nordh. Introduction to the maximum solution problem. In Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer, editors, *Complexity of Constraints – An Overview of Current Research*, volume 5250 of *LNCS*, pages 255–282. Springer, 2008.

[15] Peter Jonsson and Johan Thapper. Approximability of the maximum solution problem for certain families of algebras. In *Proceedings of CSR-2009*, pages 215–226, 2009.

[16] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6) :1863–1920, 2000.

[17] Subhash Khot. On the unique games conjecture (invited survey). *Annual IEEE Conference on Computational Complexity*, pages 99–121, 2010.

[18] Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. In *Proceedings of SODA-2012*, pages 750–759, 2012.

[19] Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of ITCS-2012*, pages 484–495, 2012.

[20] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP ? In *Proceedings of STOC-2008*, pages 245–254, 2008.

[21] Prasad Raghavendra and David Steurer. How to round any CSP. In *Proceedings of FOCS-2009*, pages 586–594, 2009.

[22] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of STOC-1978*, pages 216–226, 1978.

[23] Thomas Schiex, Helene Fargier, and Gerard Verfaillie. Valued constraint satisfaction problems : Easy and hard problems. In *Proceedings of IJCAI-1995*, pages 631–637, 1995.

[24] Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. arXiv :1204.1079 [cs.CC], 2012.